

Tampereen Ammattikorkeakoulu
Ohjelmistotekniikka
Tietotekniikka

Yhtenäisen tutkimusportaalin suunnittelu

Johan-Maurits Marsi

Työn ohjaaja
Työn tilaaja
Tampere 14.4.2011

Lehtori Erkki Hietalahti
Nordic Industries Development Oy

Tekijä	Marsi, Johan-Maurits
Työn nimi	Yhtenäisen tutkimusportaalin suunnittelu
Sivumäärä	21 sivua
Valmistumisaika	18.10.2010
Työn ohjaaja	Lehtori Erkki Hietalahti
Työn teettäjä	Nordic Industries Development Oy

TIIVISTELMÄ

Tässä työssä toteutetaan tiedonhakuportaali Nordic Industries Development Oy:lle.

Tiedonhakuportaali on web-pohjainen sivusto, johon useat käyttäjät voivat tallentaa, muokata sekä hakea erilaisia tietoja. Tiedot sisältävät linkkejä, yhteystietoja sekä artikkeleita yritystä kiinnostavista aihealueista, joita yritys saattaa tarvita uusien yhteistyökumppaneiden sekä asiakkaiden hankkimisessa.

Opinnäytetyö sisältää tiedonhakuportaalin suunnittelun, toteutuksen sekä käyttöönoton.

Suunnittelun aikana punnitaan tiedonhakuportaalin perustoimintoja sekä niiden toiminnallista rakennetta, valitaan käyttöalusta sekä –ympäristö. Toteutuksen aikana portaali toteutetaan julkaistavaan muotoon ohjelmoimalla perustoiminnot sekä niiden toivottu toiminta. Käyttöönoton aikana pystytetään palvelin, johon portaali asetetaan toimimaan toivotulla tavalla. Opinnäytetyössä tarkastellaan myös sekä analysoidaan hieman eri vaiheiden toimintaa sekä merkitystä.

Writer	Marsi, Johan-Maurits
Thesis	Designing of integral research portal
Pages	21 pages
Graduation time	18.10.2010
Thesis Supervisor	Lecturer Erkki Hietalahti
Co-operating Company	Nordic Industries Development Ltd.

ABSTRACT

This thesis is about creating integral research portal to a company Nordic Industries Development Ltd. Integral research portal is a web-based site, where multiple users can save, edit and search information. Information includes links, contact information and articles of interesting subjects, which the company might need for finding new affiliates and customers.

Thesis includes research, development and implementation. In research we design all the features and structural build and choose operating environment. In development we program the portal and all the wanted features. In implementation we choose the server and publish search portal. We also examine and analyze functioning and meaning of different features.

SISÄLLYSLUETTELO

1 Johdanto	5
2 Yleistä	6
2.1 PHP-kieli	6
2.2 MySQL-tietokanta	6
2.3 Eväste	6
2.4 Istunto.....	7
2.5 Hash (MD5) -algoritmi	7
2.6 RSS-syöte.....	8
2.7 SQL-injektio.....	8
2.8 Citrix-palvelu	9
3 Suunnittelu	10
3.1 Odotukset	10
3.2 Käyttöympäristö.....	10
4 Toteutus	11
4.1 Tietokanta.....	11
4.2 Portaali	12
4.2.1 Perustoiminnot	12
4.2.2 Käyttäjäprofiilit.....	13
4.2.3 RSS-syötteet.....	13
4.2.4 Liitetiedostot	14
4.2.5 Viimeistely	15
5 Käyttöönotto.....	16
6 Ohjelmistot.....	17
6.1 Ohjelmointityyli	17
7 Yhteenveto	18
8 Lähdeluettelo.....	19
Liite 1: Osa RSS-syötteen käsittelykoodista.....	20
Liite 2: ER-kaavio.....	21

1 Johdanto

Työn tarkoituksena oli luoda Nordic Industries Development Oy:lle yhtenäinen tutkimusportaali, johon yritys voi kerätä tietoja sekä dataa yrityksen kannalta tärkeitä tutkimuksia varten. Työssä suunnitellaan sekä toteutetaan tutkimusportaali yrityksen toiveiden sekä jo olemassa olevien tutkimusmateriaalien ympärille.

Nordic Industries Development Oy on konsultointiyritys, joka kehittää ja toteuttaa strategioita eri yritysten ulkomaankauppaa varten. Se perustettiin vuonna 1994 Suomessa. Vuonna 2000 se avasi toimistonsa Ranskaan, missä se on kehittänyt, koordinoanut sekä luonut yhteyksiä Pohjoismaisille yrityksille. Vuonna 2003 Singaporeen avattu toimisto avasi ovet Aasiaan, ja 2006 toiminta laajeni UAE-maihin.

2 Yleistä

2.1 PHP-kieli

PHP (Hypertext Preprocessor, alunperin Personal Home Page) on proseduraalinen skriptimäinen ohjelmointikieli, jota käytetään Web-palvelinympäristöissä dynaamisten web-sivujen luonnissa. PHP on monialustainen komentosarjakieli, jossa ohjelmakoodi tulkitaan vasta ohjelman suoritusvaiheessa. Se julkaistiin vuonna 1995 ja on nykyään eräs johtavista dynaamisten web-palveluiden tuottamiseen tarkoitetuista kielistä. (Wikipedia: PHP, 1)

Lähes koko portaali on ohjelmoitu käyttäen PHP-kieltä.

2.2 MySQL-tietokanta

MySQL on relaatiotietokantojen hallintajärjestelmä, joka tukee monen eri käyttäjän yhtäaikaista pääsyä palvelimella sijaitseviin tietokantoihin. MySQL toimii monissa eri käyttöjärjestelmissä ja on siitä johtuen noussut yhdeksi suosituimmista tietokantahallintajärjestelmistä. Se on hyvin yleinen etenkin web-palveluiden tietokantana. MySQL julkaistiin vuonna 1995. (Wikipedia: MySQL, 2)

Portaali käyttää MySQL-tietokantaa.

2.3 Eväste

Evästeet (engl. cookies) ovat dataa, jonka web-palvelin tallentaa käyttäjän tietokoneelle. Sanalla ”eväste” tarkoitetaan sellaista tietoa, jonka palvelun tarjoajan palvelin lähettää käyttäjän selainohjelmalle pyytäen selainta tallentamaan tiedon käyttäjän päätelaitteelle ja jota kyseinen palvelun tarjoajan palvelin voi myöhemmin pyytää takaisin. Yleisesti kyse on pienestä tietomäärästä, kuten esim. lyhyestä tekstistä. Selain lähettää tallennetut evästeet vain sille palvelimelle, joka on ne alun perin käyttäjälle lähettänyt. (Wikipedia: Eväste, 3)

Portaali käytti alunperin evästeitä, mutta ne korvattiin tarkoitukseen paremmin sopivilla istunnoilla.

2.4 Istunto

Istunto (engl. session) on tietotekniikassa käytäntö, jolla luodaan yhteys käyttäjän sekä palvelimen välille.

PHP sisältää sisäänrakennetun istuntojärjestelmän, joka yleisesti käyttää hyväkseen evästeitä, mutta osaa sijoittaa istuntotunnuksen myös jokaiseen PHP:n generoimaan hyperlinkkiin tukien näin periaatteessa myös selaimia joissa ei ole kytketty päälle tukea Javascriptille. PHP:n istunto tallettaa erikoismuuttujaan \$_SESSION kaikki koodin määrittelemät arvot, tukien näin samoja toimintoja kuin evästeet, sekä toimintoja jotka ovat arkaluontoisia kuten: salasanat, suojattujen sivujen suojatasoja ja niin edelleen. PHP istunto merkataan inaktiiviseksi selaimen viimeisen ikkunan sulkeuduttua eikä istuntoa voi tallentaa ilman ulkopuolista ohjelmaa tai tietoa (kuten selaimen istuntotallennusmekanismia tai evästettä). (Wikipedia: Ohjelmallinen Istunto, 4)

Portaali käyttää istuntoja käyttäjien hallinnassa.

2.5 Hash (MD5) -algoritmi

MD5 on niin kutsuttu Message-Digest -algoritmi, jota käytetään viestien salauksessa. Tämä tarkoittaa sitä, että MD5-algoritmi tuottaa annetusta syötteestä 128-bittisen tiivisteen, joka tyypillisesti esitetään 32-merkkisenä heksakoodatussa muodossa (esim sanan ”salasana” MD5-tiiviste on e7e941b1f09f266540c6780db51d5f58). Tämä koodaus toimii vain yksisuuntaisesti, joten kerran koodattua viestiä ei voi enää purkaa alkuperäiseen muotoonsa.

MD5 kehitettiin vuonna 1991 korvaamaan edellistä MD4-algoritmia. MD5-algoritmi kuuluu professori Ronald Rivestin suunnittelemaan Message-Digest -algoritmisarjaan. (Wikipedia: MD5, 5)

MD5-algoritmia käytetään portaalissa käyttäjien salasanojen salaamisessa.

2.6 RSS-syöte

RSS (Really Simple Syndication) on joukko verkkosyötemuotoja, joita käytetään usein päivittyvän digitaalisen sisällön julkaisemiseen (blogit, uutiset jne). Rss-syötteitä luetaan erillisillä ohjelmilla, joita kutsutaan lukijoiksi tai lukijaohjelmiksi.

RSS toimii siten, että käyttäjä tilaa syötteen lisäämällä lukijaan linkin syötteeseen, jonka lukija tarkistaa uuden sisällön varalta. Jos palvelun tarjoaja julkaisee uutta sisältöä, niin lukija hakee kyseisen sisällön ja esittää sen käyttäjälle.

RSS-tiedosto on määritetty XML-kielellä (datamuotojen yleinen määritelmä). Ensimmäiset syötteet, joiden perusidea perustui RSS-syötteisiin, kehitettiin noin vuonna 1999, vaikka idea RSS-syötteistä kehittyi joidenkin lähteiden mukaan jo v. 1995. Sen kehitti intialainen ohjelmoija Ramanathan V. Guha Netscape-selaimelle ja se tunnettiin silloin nimellä RDF Site Summary (nykyään RSS 0.9). (Wikipedia: Rss, 6)

Eräs portaalin toiminnosta on RSS-syötteiden esittäminen tarkasteltaessa yksityiskohtaisesti tallennettuja tietoja.

2.7 SQL-injektio

SQL-injektio (engl. SQL injection) on tekniikka, jolla tunkeudutaan järjestelmiin hyödyntäen tietoturva-aukkoja. Niitä esiintyy tietokantapohjaisissa sovelluksissa ja ne ovat varsin yleisiä WWW-pohjaisissa sovelluksissa, joissa käyttäjät käyttävät tietokantaa WWW-rajapinnan yli. SQL-injektiot eivät kuitenkaan ole WWW-sidonnaisia.

SQL-injektiossa hyökkääjä antaa tietokantapalvelimelle SQL-komentoja, joita hänen ei pitäisi pystyä antamaan. Tämä hyökkäys tapahtuu useimmiten puuttuvan tai väärin toteutetun syöttötiedon tarkistuksen kautta, ja joissain tapauksissa myös itse tietokantarajapinnassa tapahtuvan tiedon väärästä käsittelystä. (Wikipedia: SQL-injektio, 7)

Klassinen esimerkki:

Tavallisen web-sivustolla sijaitsevan tekstikentän tarkoitus on tarkistaa tietokannasta salasanan oikeellisuus. Tekstikenttä toimii seuraavanlaisella koodilla:

execute("SELECT * FROM Users WHERE password='" + Text + "';");

Jos tekstikenttään sijoittaisi salasanan sijasta tekstin

' OR ''='

syntyisi seuraavanlainen koodi:

SELECT * FROM Users WHERE password='' OR ''=';

Syntynyt koodi päästää hyökkääjän sisälle, ilman toimivaa salasanaa.

Testattaessa portaalin tietoturvaa apuna käytettiin SQL-injektioita.

2.8 Citrix-palvelu

Citrix Systems on monikansallinen yritys, joka tarjoaa monia web-palveluja, kuten palvelin- ja työasemavirtualisointiratkaisuja, verkkoratkaisuja, software-as-a-service (SaaS) -palveluja sekä pilvipalveluita (cloud computing). Yhtiö perustettiin vuonna 1989 ja heidän pääkonttorinsa sijaitsee Floridassa. (Wikipedia: Citrix, 9)

Citrix-palvelua käytettiin portaalin luontivaiheessa aktiivisesti.

3 Suunnittelu

3.1 Odotukset

Työn suunnittelu lähti liikkeelle siitä, mitä yritys odotti valmiilta työltä: Web-pohjaista tiedonhakuportaalia, joka sisältäisi toiminnot tiedon tallennukselle, muokkaamiselle sekä poistamiselle. Kriteeri oli se, että yrityksen senhetkiset materiaalit sekä dokumentit piti saada osaksi portaalin tiedonhakua. Valmiita ohjelmistoja ei ollut, joten kyseinen portaali tulisi ohjelmoida puhtaalta pöydältä.

Monipuolisuus, yksinkertaisuus, turvallisuus sekä toimivuus eri internet-selaimissa nousivat tärkeiksi prioriteeteiksi. Myös se, että sivustoa pystyisi käyttämään yhtäaikaaisesti mahdollisimman moni työntekijä, nousi vaatimukseksi.

Yritys toivoi, että sivuston valmistuttua se olisi integroitu osa yrityksen nykyistä nettisivustoa, joka on osittain toteutettu ASP Classic-kielellä.

3.2 Käyttöympäristö

Koska sivusto tulisi sisältämään paljon muuttuvia tietoja, oli ainoa ratkaisu luoda tietokanta niitä varten. Tietokannaksi valittiin MySQL hinnan, laajan tuen sekä sen yleisyyden johdosta.

Suunnitelmat sivuston luomiseksi ASP Classic-kielellä piti hylätä jo projektin alkuvaiheessa. Hylkäämisen syynä oli allekirjoittaneen puutteelliset tiedot sekä taidot kyseisen kielen toiminnasta sekä syntaksista. Kielen opiskelu ei kannattanut, koska kyseinen kieli osoittautui liian työlääksi opetella annettua aikarajaa silmällä pitäen. Myös ASP Classic -kielen Microsoft-tausta olisi mahdollisesti tuottanut ongelmia, kun sivustoa olisi testattu eri käyttöympäristöissä, kuten Macintoshissa sekä Linuxissa.

Koska työntekijöiden omien tietokoneiden asetukset saattavat erota toisistaan, niin kieleksi valittiin PHP sen yleisyyden sekä toimivuuden johdosta. Muut kielet (javascript, flash jne) jätettiin pois, koska ne vaativat toimiaksensa jonkin erikseen asennettavan lisäosan, tai kieli muuten vain rajoitti eri toimintojen käyttöä. PHP mahdollistaa myös tietokannan monipuolisen käsittelyn, joka helpottaa tulevan portaalin rakentamista.

4 Toteutus

Portaali luotiin kaksivaiheisesti: Aluksi luotiin itse tietokanta ja tämän mukaan suunniteltiin itse portaali. Tämä kohtalaisen nopeasti tehty ratkaisu sisälsi kauaskantoisia vaikutuksia portaalin suunnittelun sekä toteutuksen kannalta.

4.1 Tietokanta

Aluksi yrityksen alkuperäiset excel-taulukot muokattiin yhdeksi yhtenäiseksi taulukoksi, joka sitten siirrettiin MySQL-tietokantaan. Excel-taulukoiden yhdistäminen oli yksinkertainen mutta aikaavievä prosessi, koska eri taulukoiden rakenteet poikkesivat toisistaan sisällöllisesti. Eri tietojen merkitystä ja tarvetta punnittiin ja lopuksi päädyttiin siihen, että kaikki tieto oli tarpeellista säilyttää eikä mitään tarvinnut karsia pois. Joidenkin sarakkeiden nimiä piti vaihtaa sekä yhdenmukaistaa, jotta jonkinlainen yksinkertaisuus sekä selkeys säilyisi. Tämä vaati tietojen paloittelua eri osioiksi ja lopuksi näiden eri osioiden yhdistämistä yhdeksi kokonaiseksi tiedostoksi. Tätä vaihetta ei voitu automatisoida, koska tieto oli hajautettua ja usean eri henkilön suunnittelema.

Testitietokanta sijoitettiin koulun tarjoamalle palvelimelle, koska se oli heti käyttövalmis sekä mahdollisti etäkäytön. Taulukon siirto testitietokantaan kävi kohtalaisen nopeasti MySQL-komentojen avulla. Koska testitietokanta sijaitsi koulun tietokannassa, tuottivat tiedostopolut pientä pääänvaivaa taulukoiden siirrossa.

Taulukon siirto palvelimelle onnistui lopulta seuraavalla käskyllä:

```
LOAD DATA LOCAL INFILE '/mnt/st/t/[omatunnus]/sql.csv' INTO TABLE taulu FIELDS  
TERMINATED BY ';' LINES TERMINATED BY '\r\n';
```

Vaikka testitietokanta sisälsi kaikki excel-taulukoiden tiedot, se ei kuitenkaan ollut vielä valmis.

Liian hätiköidystä suunnittelusta johtuen tietokanta vaati lähes jatkuvaa muutosta aina palvelimen pystyttämiseen asti: Tietokantaan piti lisätä jälkikäteen muutamia uusia tietueita (esim yksilölliset ID-numerot) sekä tietokannan rakennetta piti muuttaa näiden uusien tietueiden vuoksi.

Sivuston sisällön laajentuessa tuli tarve luoda yksilölliset käyttäjäprofiilit, käyttäjäprofiilien hallinta sekä tiedostojen hallinta liitetiedostoja varten. Tästä johtuen tietokantaan lisättiin erillisiä tauluja kyseisiä toimintoja varten, jotka sisältävät käyttäjätiedot sekä liitetiedostojen linkitykset niihin kuuluviin tietoihin.

Tietokannan sekä sivuston valmistuttua ne siirrettiin hankitulle palvelimelle. Tietokannan siirtäminen palvelimelle kävi nopeasti sen jälkeen, kun palvelimen ympäristö oli saatu asennettua loppuun.

4.2 Portaali

Portaalin testiversio luotiin koulun jokaiselle oppilaalle tarjoamalle 300MB:n public-kansiolle, koska tämä oli nopea ja valmis ratkaisu eikä vaatinut erillisten testipalvelimien asentamista. Lähestymistapa oli se, että aina uuden toiminnon valmistuessa aloitetaan toteuttamaan seuraavaa osaa. Kaikkien toimintojen yhäaikainen toteuttaminen ei onnistunut, koska sivuston luontivaiheessa ei ollut täysin selvää mitä toimintoja sivun kuului lopulta sisältää.

4.2.1 Perustoiminnot

Portaalin toteutus lähti liikkeelle perustoimintojen ohjelmoimisella. Aluksi sivusto koostui kahdesta sivusta, etusivusta sekä tietokannan käsittelijästä, jotta toteutus ei lähtisi paisumaan liikaa. Käsittelijä koostui yksinkertaisista toiminnoista: muutamien tietokenttien tallentamisesta tietokantaan sekä tietokannan sisällön näyttämisestä selaimessa. Nämä perustoiminnot tulisivat olemaan sivuston selkäranka, joten niiden saaminen toimintavalmiiksi oli aluksi tärkeintä.

Kun nämä toiminnot oli saatu tehtyä valmiiksi, niin alkoi seuraavien osien toteutus. Samalla päivitettiin valmiiden sivujen koodia, jotta ne toimisivat uusien sivujen kanssa.

Tietokantaan lisääminen, muokkaaminen sekä poistaminen valmistuivat nopeasti. Kyseiset toiminnot olivat kuitenkin hyvin alkeellisia, jotka vaativat päivittämistä monipuolisemmaksi. Tiedon esittäminen tuotti myös pieniä haasteita: Yksi tietue kun sisältää 13 eri tekstikenttää (esim nimi, maa, toimenkuva), joista useimmat pystyvät sisältämään jopa satoja merkkejä. Tämän tiedon rajaaminen ja esitystyylit toi tästä syystä ylimääräisiä haasteita: mitkä tiedot ovat ensisijaisen tärkeitä haun kannalta ja mitkä luokitellaan vähemmän tärkeiksi tietoja hakiessa.

Samalla kun perustoimintojen toimivuus parani, tuli tarve salata sivujen toiminnot ulkopuolisilta tahoilta. Tätä varten sivuille luotiin kirjautumisikkuna sekä tietokantaan uusi taulu käyttäjäprofiileja varten. Tämä vaihe ei kuitenkaan ollut yksinkertainen lisäys, vaan piti punnita eri kirjautumistapojen toiminnallisuuksia sekä mahdollisuuksia. Aluksi luotiin etusivulle tekstipalkit sisäänkirjautumista varten, jonka jälkeen jokaiselle sivulle luotiin pieni koodinpätkä, joka varmisti että onko käyttäjä kirjautunut sivulle vai ei.

4.2.2 Käyttäjäprofiilit

Ensimmäinen kirjautumistoiminto perustui evästeisiin. Tämä tarkoitti sitä, että käyttäjän kirjautuessa sisälle sivusto luo evästeen, josta sivusto tarkistaa käyttäjän tilan. Tämä ratkaisu toimi niin kuin piti, mutta herätti tietoturvakysymyksiä hieman valmistumisensa jälkeen: käyttäjän tunnus sekä salasana tallentuivat evästeeseen, jolloin ulkopuolisen olisi helppo tarkistaa nämä selaimen välimuistista. Salasanan salaaminen ei ollut vartenotettava vaihtoehto, koska evästeen kopioiminen toiseen selaimen olisi mahdollisesti kiertänyt tietoturvan. Eväste-pohjainen sisäänkirjautuminen korvattiin istunto-pohjaisella sisäänkirjautumisella. Koska kirjautumistiedot sijaitsevat nyt palvelimella, on tietoturva parempi kuin evästeitä käytettäessä. Istunto-pohjainen sisäänkirjautuminen oli helppo tehdä, koska jokaiselle sivulle luotiin pieni tarkistuskoodi, joka tarkistaisi istunnon senhetkisen tilan. Tämä vaihtoehto toimi paremmin kuin evästeet, joten se jäi pysyväksi ratkaisuksi.

Tämä kuitenkin loi aivan uudenlaisen tarpeen käyttäjäprofiilien hallinnalle. Tämä toiminto oli lähes kokonaan sivuutettu alkuperäisessä suunnitelmassa, joten se vaati nopeita ratkaisuja. Millainen hallintaikkunasta pitäisi tulla ja keillä olisi oikeus käyttää sitä? Muutaman päivän suunnittelun jälkeen päädyttiin siihen, että etusivulle luodaan linkki käyttäjähallintaan. Tämä linkki on nähtävillä ainoastaan ylläpitäjille. Jos peruskäyttäjä sattuisi saamaan kyseisen linkin haltuunsa, niin sivusto ohjaisi käyttäjän takaisin etusivulle tarkistettuaan käyttäjän valtuudet. Vain ylläpitäjillä on mahdollisuus luoda, muokata sekä poistaa käyttäjäprofiileja. Muilla käyttäjillä on vain oikeus vaihtaa salasanansa.

4.2.3 RSS-syötteet

RSS-syötteet ovat nykyään yleinen tapa kertoa uusimmista uutisista sekä tapahtumista, joten sivuilla tulisi olla mahdollisuus tarkastella halutun sivuston RSS-syötteitä. Tavallisesti nämä

syötteet näkyvät selaimen RSS-lukijassa sen jälkeen, kun sivuston osoite on lisätty selaimen RSS-listalle. Tämä ei kuitenkaan ole toimiva ratkaisu portaalia silmällä pitäen, koska muuten jokaisen käyttäjän koneelle pitäisi asentaa kaikkien mahdollisten sivujen syötteet. Tämä aiheuttaisi tietotulvan. Portaaliin piti rakentaa erillinen ikkuna, jossa näytetään haettavan sivuston mahdollinen RSS-syöte ilman mitään erillisiä ohjelmistoasennuksia. RSS-syötteen haku eri sivuilta tuotti päänsäivä, koska vastoin yleistä standardia useiden sivujen lähdekoodissa viitataan virheellisesti syötteisiin ja niiden sijaintiin. Tällöin tavalliset syötteenlukijat eivät pysty havaitsemaan näitä, vaikka syötteet olisivatkin tarjolla. Myös eri selainten RSS-tuki vaihteli merkittävästi (esim Google Chrome ei tukenut toteutusvaiheessa RSS:ää yhtä tehokkaasti kuin muut selaimet). Muutamien kokeilujen sekä kompromissien jälkeen Search Portaliin saatiin asennettua pienimuotoinen RSS-lukija. Kyseinen RSS-lukija näyttää pienessä kehyksessä haettujen sivujen viimeisimmät syötteet. Se pystyy havaitsemaan sivujen lähdekoodista RSS-syötteet vaikka kyseinen syötekoodi ei olisikaan täysin sääntöjen mukainen. Täysin mielivaltaisesta syötekoodista ei kuitenkaan pystytä poimimaan syötettä, vaan koodin perusrunko tulee olla sama kuin yleinen RSS-standardi sen määrittää, jotta syöte pystytään havaitsemaan (Kuva 1).

Oikeaoppinen viittaus RSS-syötteeseen sivuston lähdekoodissa (RSS Autodiscovery, 9):

```
<link rel="alternate" type="application/rss+xml" title="otsikko"
href="http://www.esimerkki.fi"/>
```

Tiedonhakuportaalien RSS-lukija vaatii toimiaksensa seuraavat tunnisteet kyseisestä viittauksesta:

- 1) **type**-tunniste, joka kertoo kyseessä olevan RSS-syöte
- 2) **href**-tunniste, joka kertoo RSS-syötteen sijainnin

rel- sekä **title**-tunnisteita ei huomioida, joten niiden puuttuminen ei vaikuta portaalien RSS-lukijan toimintaan.

Osa RSS-syötteen käsittelykoodista löytyy esimerkkinä liitteessä 1.

4.2.4 Liitetiedostot

Pelkkä tekstin tallentaminen ei yksistään riitä, koska se toimisi rajoittavana tekijänä tiedon monipuolisuudelle. Tämän johdosta tuli tarve saada mahdollisuus liittää tiedostoja sekä dokumentteja liitteiksi. Liitteiden tiedostomuoto oli aluksi avoin. Tiedostomuodon määrittäminen yhdenlaiseksi ei käynyt, koska se olisi rajoittanut liikaa käyttäjiä sekä se olisi vaatinut käyttäjiä

konvertoimaan kelpaamattomat tiedostot oikeaan muotoon. Tämä olisi vaivalloista ja heikentäisi sivuston käyttäjäystävällisyyttä. Kaikkien tiedostomuotojen hyväksyminenään ei ollut hyvä vaihtoehto, koska tällöin käyttäjillä olisi mahdollisuus ladata tulevalle palvelimelle esim. haittaohjelmia. Myös kokorajoitus on välttämätön, koska liian isot tiedostot täyttäisivät palvelimen rajoitetun kapasiteetin hyvinkin nopeasti.

Tiedostojen muoto rajoitettiin yleisimpiin tiedostomuotoihin, kuten esim. PDF-, DOC- ja TXT-tiedostoihin sekä JPEG- että PNG-kuviin. Kokorajoitus määritettiin yrityksen nykyisten tiedostokokojen perusteella. Tiedostoja varten piti myös luoda tietokantaan uusi taulu, koska tiedostolinkkien tallennus päättietokantaan olisi tehnyt siitä sekavan sekä hankalan ylläpitää.

Pelkkä tiedostojen tallentaminen ei kuitenkaan ole riittävä toiminto, koska käyttäjillä tulisi olla mahdollisuus poistaa sekä lisätä jälkeensä uusia tiedostoja. Näiden toimintojen lisääminen vaati muutoksia muutamissa sivuissa, mutta lopulta tiedostojen tallennus sekä poistaminen toimi niin kuin pitikin. Nyt kuka tahansa käyttäjä (jolla on tunnukset) voi lisätä sekä poistaa tiedostoja. Tiedostojen poistoa ei voitu rajoittaa vain ylläpitäjien oikeudeksi, koska tämä olisi vaatinut käyttäjiltä vielä tiiviimpää yhteydenpitoa ylläpitäjien kanssa, joka taas olisi hidastanut ylläpitäjien normaalia työntekoa.

4.2.5 Viimeistely

Sivuston lähestyessä julkaistavaksi kelpaavaa beta-versiota, heräsi tarve testata sivuston tietoturvallisuutta. Testatessa sivustoa tavallisella SQL-injektionilla tuli esille sivun haavoittuvuus: tekstikentät eivät suodattaneet laisinkaan annettujen tietojen sisältöä, vaan sivusto käsitteli annetun tiedon kuin osana sivuston lähdekoodia. Suodattamalla tekstikenttien sisällön epätavallisilta merkeiltä väheni riski SQL-injektioiden toimivuudesta. Samalla tuli testatuksi sivustojen yleinen tietoturvasato.

5 Käyttöönotto

Portaalin tultua julkaisuvaiheeseen se piti siirtää palvelimelle ja testata sen toimivuus. MySQL:n asennuksen jälkeen tietokannan siirtäminen sinne kävi nopeasti eikä suurempia ongelmia ilmennyt. Portaalin käyttöönoton aikana ilmeni ongelmia, jotka koskivat pääasiassa käyttäjien oikeuksia muuttaa kansion sisällä olevia tiedostoja. Myös ladattavien tiedostojen koko oli pielessä, koska PHP:n default-asetukset olivat odotettua pienemmät. Näitä pieniä haittoja lukuunottamatta portaali toimi niin kuin testipalvelimella. Virhekorjausten sekä muutamien lisäysten jälkeen sivusto saavutti toivotun lopputuloksen: tiedonhakuportaanin, johon valitut käyttäjät voivat tallentaa yrityksen kannalta tarpeellisia tietoja sekä tiedostoja, sekä muokata niitä tarvittaessa.

6 Ohjelmistot

Portaali luotiin täysin Microsoft Windowsin Notepad-tekstityökalulla. Syy tähän oli se, että testipalvelin (missä koko ohjelmointi tapahtui) sijaitsi koulun palvelimella, johon allekirjoittaneella ei ole muutosoikeuksia. Tämän johdosta testipalvelimelle ei voitu asentaa erillisiä ohjelmistoja, joita allekirjoittanut olisi tarvinnut työskentelyssä. Testipalvelimelle oli asennettu Notepadin kehittyneempi versio Notepad++, mutta allekirjoittanut huomasi tämän liian myöhäisessä vaiheessa. Portaalin luominen paikallisena projektina (esim. yrityksen tietokoneelle) olisi osoittanut hankalaksi etätyöskentelyä silmällä pitäen, joten tämän johdosta pidättäydyttiin koulun tarjoamalla palvelintilalla.

Ohjelmointi tapahtui yrityksen tietokoneella, joka oli Citrixin avulla yhteydessä koulun palvelimeen.

6.1 Ohjelmointityyli

Koodityyli pidettiin mahdollisimman selvänä ja helppolukuisena, jotta koodiin ennestään tutustumaton ymmärtää mahdollisimman helposti koodin toimintaperiaatteen. Koodin toimintaa on selvennetty sekä kommentoitu lähdekoodissa, jotta toiminta ei jäisi epäselväksi. Koodin ulkoasu perustuu Fredrik Kristiansenin kirjoittamaan PHP Coding Standard-tyylioppaaseen (PHP Coding Standard, 10), mutta muutamia vapautuksia on otettu esim. muuttujien nimeämisessä, joten täysin puhtasoppisesta koodista ei ole kyse.

7 Yhteenveto

Vähän yli 3 kuukauden ohjelmoinnin tulos oli tiedonhakuportaali, joka sisältää käyttäjäprofiilit sekä niiden hallinnan, tiedon lisäämisen, tarkastelun, muokkaamisen sekä poistamisen, sekä liitetiedostojen lisäämisen, muokkaamisen sekä poistamisen.

Portaalin toiminta on testattu kolmessa eri selaimessa (Mozilla Firefox, Google Chrome sekä Internet Explorer) ja varmistettu näin sen toimivuus suurimmalla osalla internetin käyttäjistä (näiden kolmen selaimen osuus kaikkien web-selainten käyttäjistä oli 86% Huhtikuussa 2011). (Usage Share of Web Browsers, 11)

Portaalin toimintaa eri käyttöjärjestelmissä ei ole testattu, mutta koska suurin osa sivuston toiminnasta pohjautuu internet-selaimen toimintaan, niin eri käyttöjärjestelmillä ei pitäisi olla suurta vaikutusta käyttäjäkohtaisen kokemuksen kannalta.

Portaali koostuu 16 php-sivusta, ja ne koostuvat yli 2600 rivistä koodia. Portaalissa käytetty PHP-versio on 5.2.6-1 ja MySQL-tietokannan versio on 5.1.25. Portaalin rakenne selviää ER-kaaviosta, joka on nähtävillä liitteessä 2.

8 Lähdeluettelo

- (1) Wikipedia 2011. PHP. [online] [viitattu 11.4.2011]. <http://fi.wikipedia.org/wiki/PHP>
- (2) Wikipedia 2011. MySQL. [online] [viitattu 11.4.2011]. <http://en.wikipedia.org/wiki/MySQL>
- (3) Wikipedia 2011. Evästeet. [online] [viitattu 11.4.2011].
<http://fi.wikipedia.org/wiki/Ev%C3%A4ste>
- (4) Wikipedia 2011. Istunto. [online] [viitattu 11.4.2011]. <http://fi.wikipedia.org/wiki/Istunto>
- (5) Wikipedia 2011. MD5-algoritmi. [online] [viitattu 11.4.2011]. <http://fi.wikipedia.org/wiki/MD5>
- (6) Wikipedia 2011. RSS-syöte. [online] [viitattu 11.4.2011]. <http://fi.wikipedia.org/wiki/Rss>
- (7) Wikipedia 2011. SQL-injektio. [online] [viitattu 11.4.2011]. <http://fi.wikipedia.org/wiki/SQL-injektio>
- (8) Wikipedia 2011. Citrix. [online] [viitattu 11.4.2011]. <http://fi.wikipedia.org/wiki/Citrix>
- (9) Cadenhead, Holderness, Morin 2006. RSS Autodiscovery. [online] [viitattu 11.4.2011].
<http://www.rssboard.org/rss-autodiscovery>
- (10) Kristiansen, Fredrik 2003. PHP Coding Standard. [online] [viitattu 11.4.2011].
<http://www.dagbladet.no/development/phpcodingstandard/>
- (11) Wikipedia 2011. Usage Share of Web Browsers. [online] [viitattu 11.4.2011].
http://en.wikipedia.org/wiki/Usage_share_of_web_browsers#Summary_table

Liite 1: Osa RSS-syötteen käsittelykoodista

```
if($found == false)
{
    $code="";
    echo "Couldn't find rss";
}
else if($found != false)
{
    //saves 600 characters long string, which will contain rss-link
    $code = substr($code,$found-300, 600);
    //we will separate the first part
    $first = substr($code,0,300);
    //and the second part into own parameters
    $second = substr($code,300);

    //this finds the last "<" -character and saves everything after that
    $first_pos = stripos($first,"<");
    $first = substr($first,$first_pos);

    //this finds the first ">" -character and saves everything before that
    $second_pos = strpos($second,">");
    $second = substr($second,0,$second_pos);

    //let's combine those
    $code = $first.$second;
    $code = strstr($code,"href=\""); //prints after first href= "

    $code = substr($code,6); //prints www.etc

    $found = strpos($code, "\""); //finds "

    $code = substr($code, 0, $found); //prints everything to the symbol "

    //now we have correct link. If the first symbol is "/", then
    //we will add original weblink to the beginning
    if($code[0]=="/")
    {
        $code = $_POST["website"].$code;
    }

    echo "Tip: Click link with scroll button to open it in a new tab.</td></tr><tr><td>";

    echo "<iframe src=\"".$code."\" id='rsslink' style='height:350;
width:700;\"></iframe></tr></td></table>";
}
```

Liite 2: ER-kaavio

